Final Proposal
INF 551

Using Cloud Database to Manage a Kaggle Dataset

Home Medical Visits

Keerti Bhogaraju
Kevin Chiang

TABLE OF CONTENTS

Introduction

Collection of the visits of a Home Medical Service Company for two years

This information is of a Home Medical Services Company located in the metropolitan area of Barcelona(Spain). The goal is to predict the level of sanitary actions in geographical areas based on environmental agents and their effect on "Fragile" people. The objective of this challenge is to be able to determine in advance the level of burden that health services will have in a given geographical area depending on environmental agents (climate, pollution, etc). It is understood by people fragile, elderly people living alone, or people with specific pathologies.

Project Idea

Building a Web application provides information about visits of a Home Medical Service Company

(1) User can obtain patient information like age, patient location, visit information done by the medical service, medical details for each visit

(2) User can also use sorting and filtering to narrow the search

(3) Providing data visualization that helps the user to get information more efficiently and do further analysis

Dataset Details

Number of rows:- 40,079

Number of columns:- 15

Size of the dataset:- 5.8MB

Non-numerical variables:- Date, Pathology, City

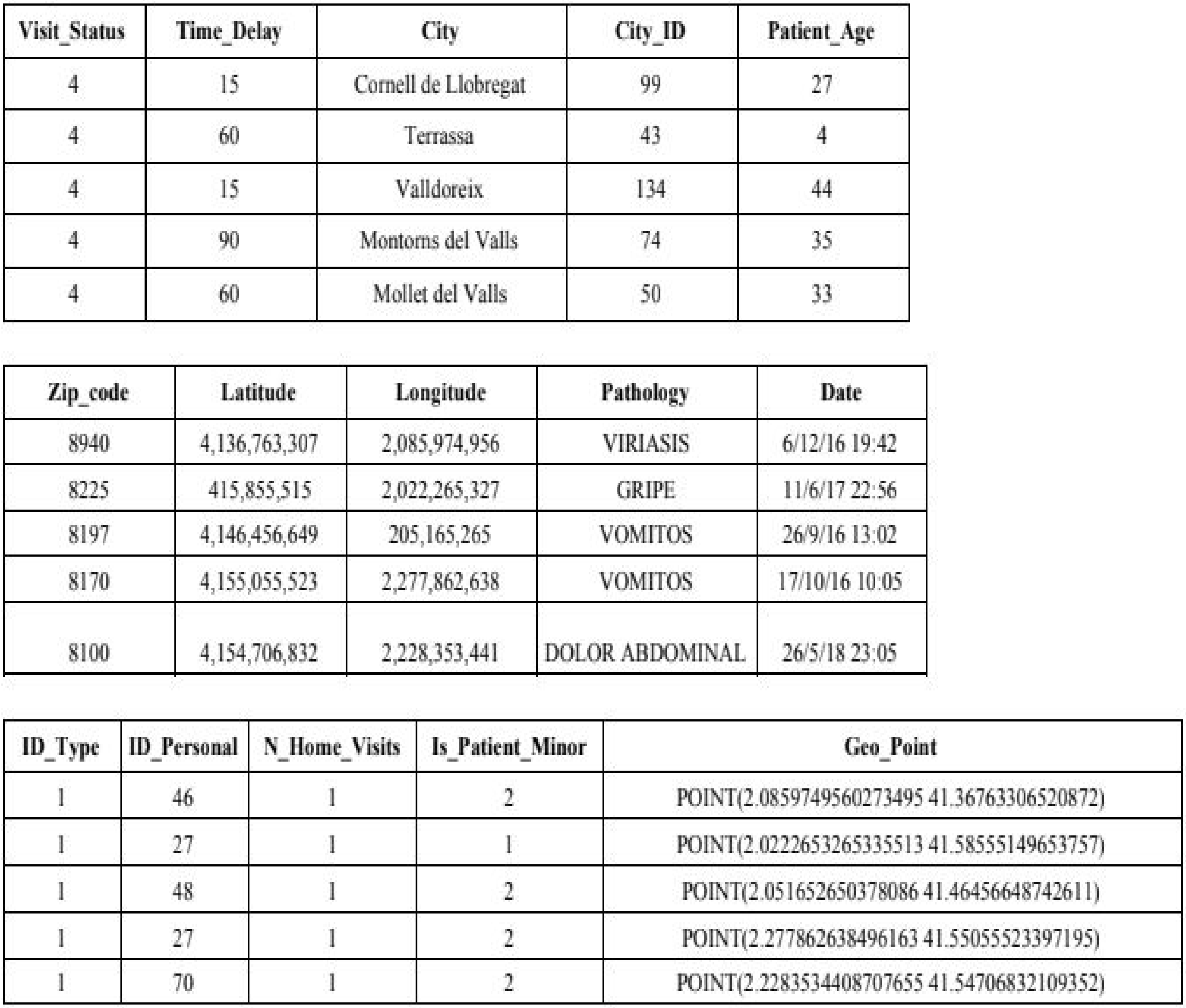
Numerical variables:- City_ID, Latitude, Longitude, N_Home_Visits, Time Delay, Date, Age, Visit_Status, ID_Type, ID_Personal, Is_Patient_Minor, Geopoint

(We do not show Geopoint on this app because it is redundant, Geopoint=Latitude&Longitude)

Data source: -

https://www.kaggle.com/HackandHealth/home-medical-visits-healthcare#Hack2018_ES.csv

Sample



Visit_Status	Time_Delay	City	City_ID	Patient_Age
4	15	Cornell de Llobregat	99	27
4	60	Terrassa	43	4
4	15	Valldoreix	134	44
4	90	Montorns del Valls	74	35
4	60	Mollet del Valls	50	33

Zip_code	Latitude	Longitude	Pathology	Date
8940	4,136,763,307	2,085,974,956	VIRIASIS	6/12/16 19:42
8225	415,855,515	2,022,265,327	GRIPE	11/6/17 22:56
8197	4,146,456,649	205,165,265	VOMITOS	26/9/16 13:02
8170	4,155,055,523	2,277,862,638	VOMITOS	17/10/16 10:05
8100	4,154,706,832	2,228,353,441	DOLOR ABDOMINAL	26/5/18 23:05

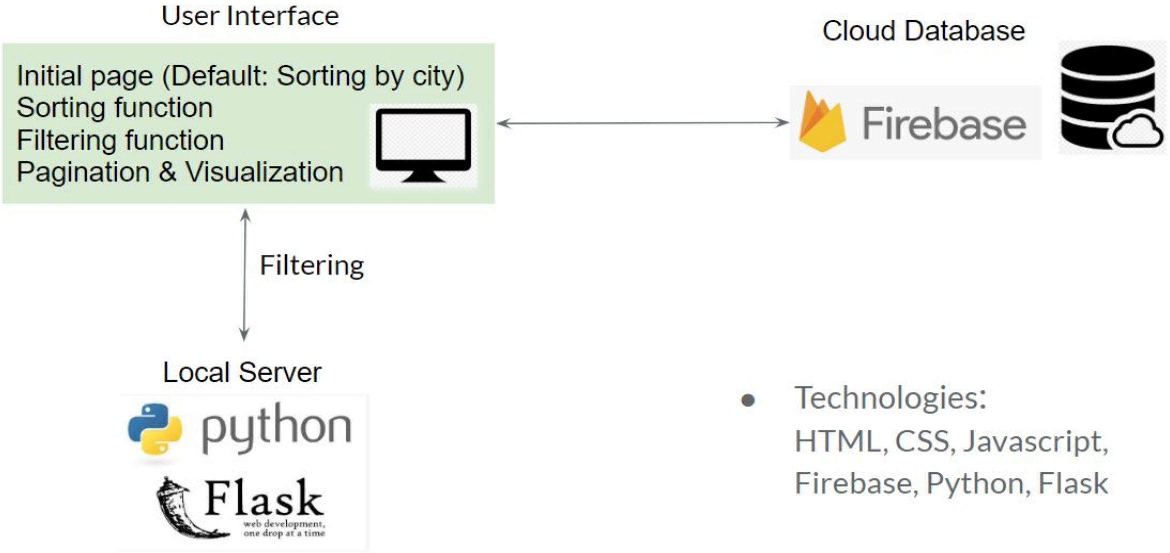
ID_Type	ID_Personal	N_Home_Visits	Is_Patient_Minor	Geo_Point
1	46	1	2	POINT(2.0859749560273495 41.36763306520872)
1	27	1	1	POINT(2.0222653265335513 41.58555149653757)
1	48	1	2	POINT(2.051652650378086 41.46456648742611)
1	27	1	2	POINT(2.277862638496163 41.55055523397195)
1	70	1	2	POINT(2.2283534408707655 41.54706832109352)

Functionalities

The following functionalities have been implemented in the Web Application -

1.) Sorting on numerical and non-numerical columns
2.) Filtering on numerical and non-numerical columns
3.) Pagination
4.) Data Visualisation

Architecture



Web Application Architecture

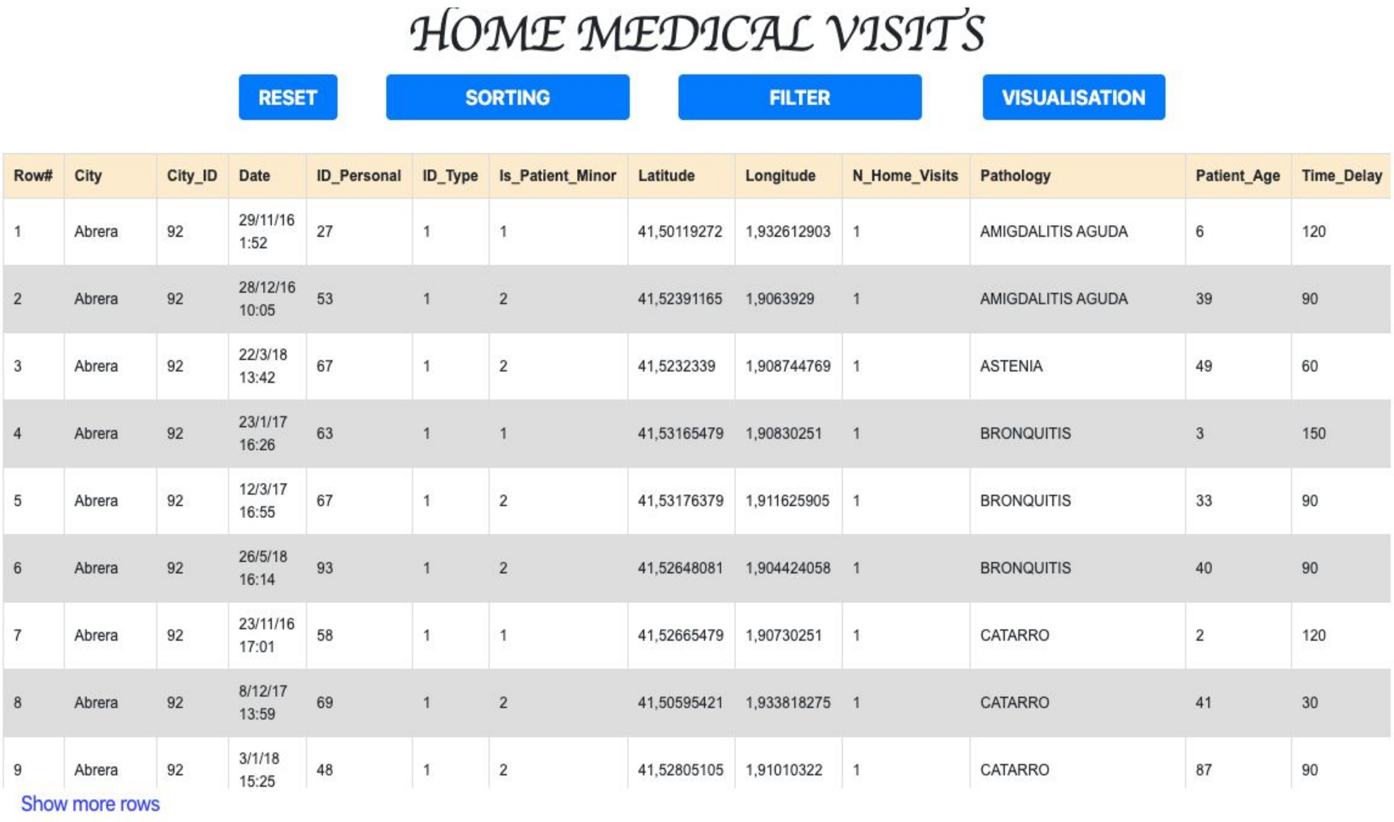
Tools and Technologies

1. Database storage - Firebase
2. Frontend display - HTML, CSS
3. Frontend operations (visualisation) - Chart.js
4. Backend operations (sorting) - Javascript
5. Backend operations (filtering) - Python Flask

Implementation Details

a.) Featuring the dataset on the website

The dataset is uploaded to Firebase using a python script. It is then fetched as a Javascript array and displayed in an HTML table. The table is sorted by non-numerical column **City**, ascending order by default.

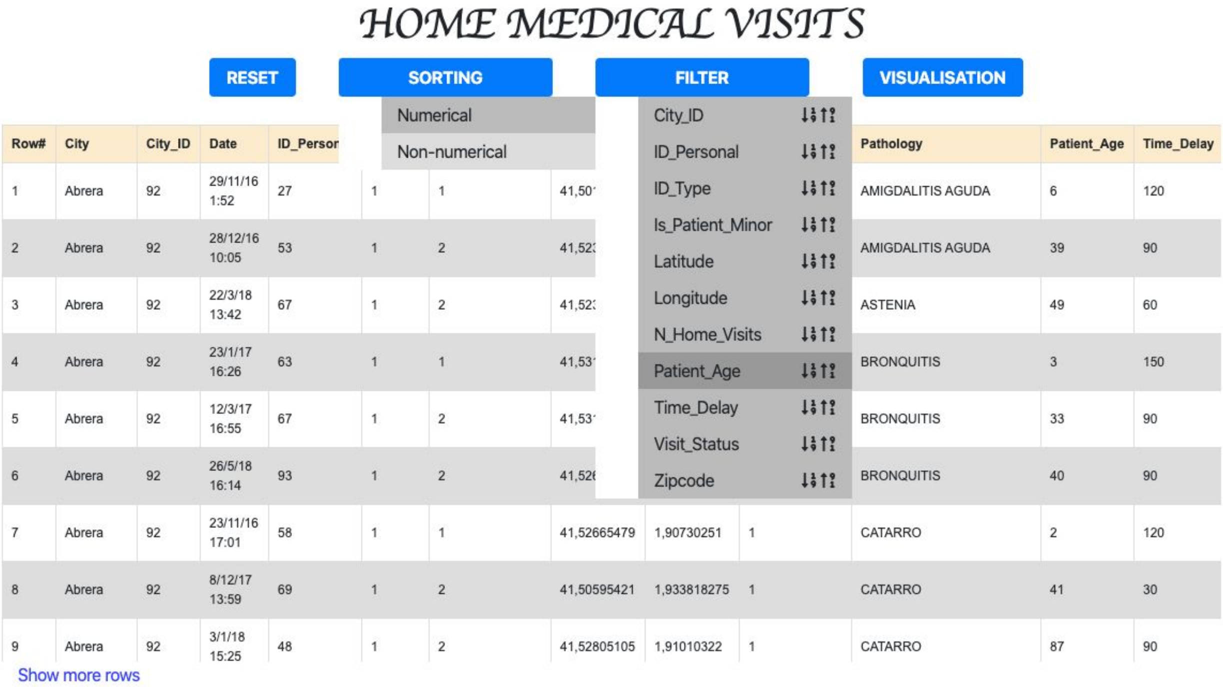


Row#	City	City_ID	Date	ID_Personal	ID_Type	Is_Patient_Minor	Latitude	Longitude	N_Home_Visits	Pathology	Patient_Age	Time_Delay
1	Abrera	92	29/11/16 1:52	27	1	1	41,50119272	1,932612903	1	AMIGDALITIS AGUDA	6	120
2	Abrera	92	28/12/16 10:05	53	1	2	41,52391165	1,9063929	1	AMIGDALITIS AGUDA	39	90
3	Abrera	92	22/3/18 13:42	67	1	2	41,5232339	1,908744769	1	ASTENIA	49	60
4	Abrera	92	23/1/17 16:26	63	1	1	41,53165479	1,90830251	1	BRONQUITIS	3	150
5	Abrera	92	12/3/17 16:55	67	1	2	41,53176379	1,911625905	1	BRONQUITIS	33	90
6	Abrera	92	26/5/18 16:14	93	1	2	41,52648081	1,904424058	1	BRONQUITIS	40	90
7	Abrera	92	23/11/16 17:01	58	1	1	41,52665479	1,90730251	1	CATARRO	2	120
8	Abrera	92	8/12/17 13:59	69	1	2	41,50595421	1,933818275	1	CATARRO	41	30
9	Abrera	92	3/1/18 15:25	48	1	2	41,52805105	1,91010322	1	CATARRO	87	90

Landing Page

b.) Sorting by numerical column, descending order

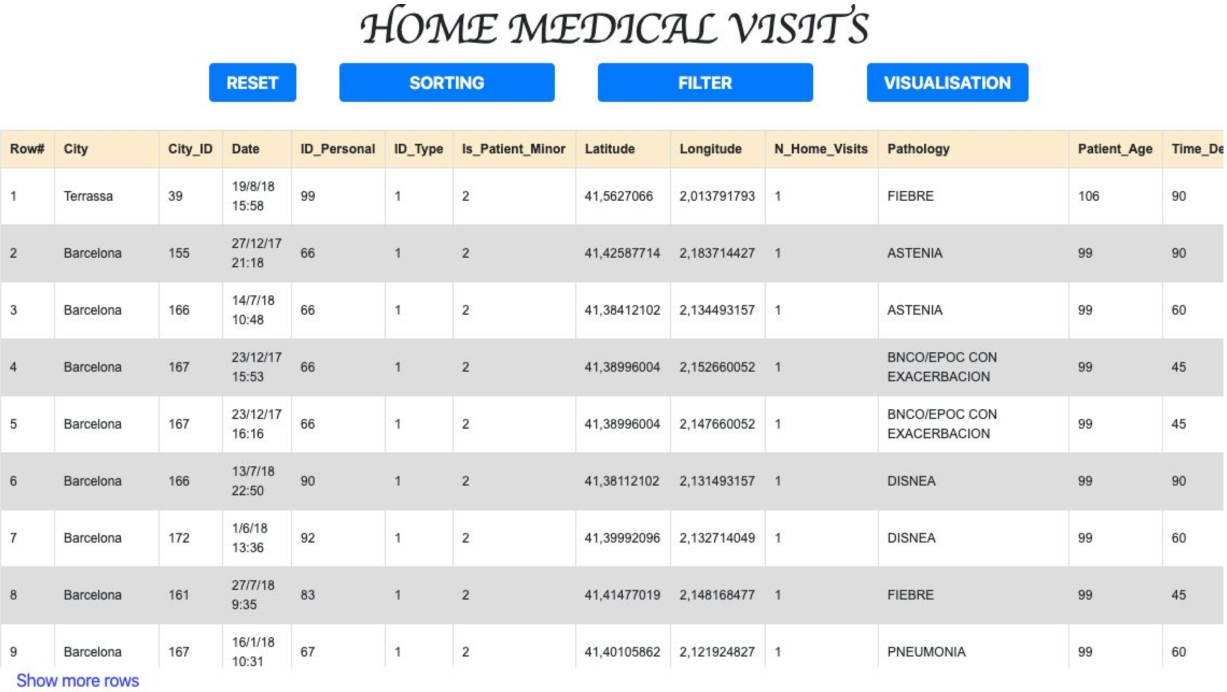
The data are stored on user's browser as Javascript array. Javascript takes care of both ascending and descending sorting. User can do multiple sorting because Javascript maintains the previous result.



HOME MEDICAL VISITS

Row#	City	City_ID	Date	ID_Persor						Pathology	Patient_Age	Time_Delay
1	Abrera	92	29/11/16 1:52	27	1	1	41,50			AMIGDALITIS AGUDA	6	120
2	Abrera	92	28/12/16 10:05	53	1	2	41,52			AMIGDALITIS AGUDA	39	90
3	Abrera	92	22/3/18 13:42	67	1	2	41,52			ASTENIA	49	60
4	Abrera	92	23/1/17 16:26	63	1	1	41,53			BRONQUITIS	3	150
5	Abrera	92	12/3/17 16:55	67	1	2	41,53			BRONQUITIS	33	90
6	Abrera	92	26/5/18 16:14	93	1	2	41,52			BRONQUITIS	40	90
7	Abrera	92	23/11/16 17:01	58	1	1	41,52665479	1,90730251	1	CATARRO	2	120
8	Abrera	92	8/12/17 13:59	69	1	2	41,50595421	1,933818275	1	CATARRO	41	30
9	Abrera	92	3/1/18 15:25	48	1	2	41,52805105	1,91010322	1	CATARRO	87	90

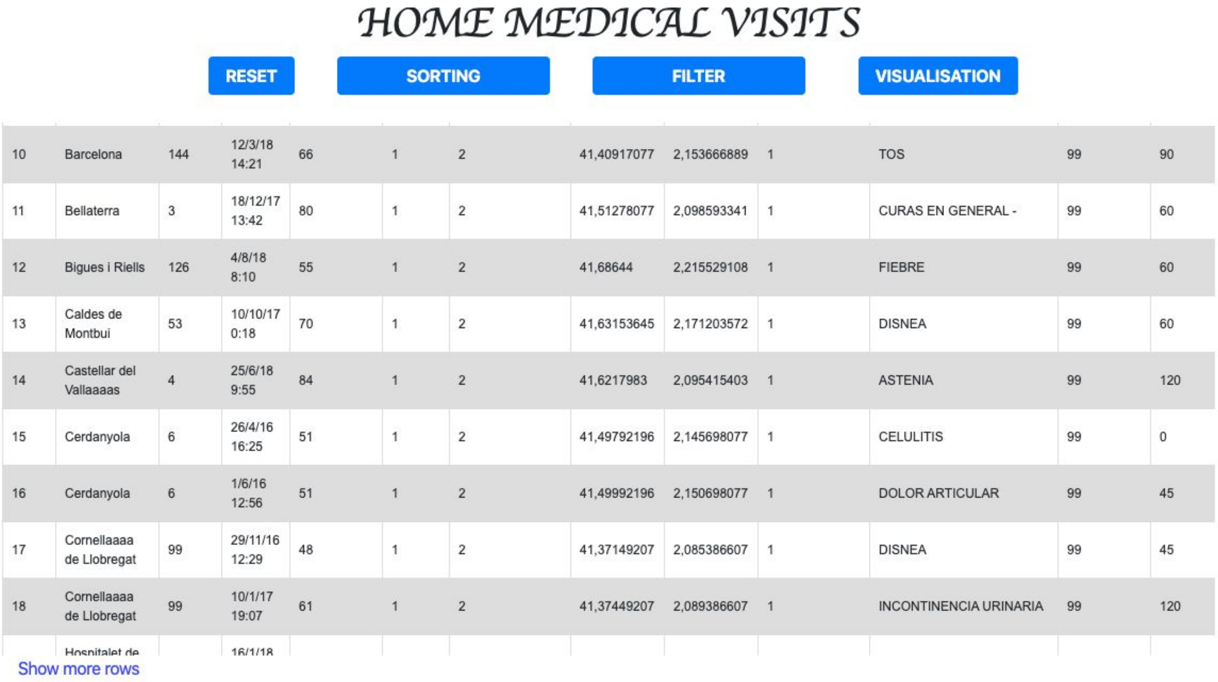
*Sorting by numerical column **Patient Age** - 1*



HOME MEDICAL VISITS

Row#	City	City_ID	Date	ID_Personal	ID_Type	Is_Patient_Minor	Latitude	Longitude	N_Home_Visits	Pathology	Patient_Age	Time_De
1	Terrassa	39	19/8/18 15:58	99	1	2	41,5627066	2,013791793	1	FIEBRE	106	90
2	Barcelona	155	27/12/17 21:18	66	1	2	41,42587714	2,183714427	1	ASTENIA	99	90
3	Barcelona	166	14/7/18 10:48	66	1	2	41,38412102	2,134493157	1	ASTENIA	99	60
4	Barcelona	167	23/12/17 15:53	66	1	2	41,38996004	2,152660052	1	BNCO/EPOC CON EXACERBACION	99	45
5	Barcelona	167	23/12/17 16:16	66	1	2	41,38996004	2,147660052	1	BNCO/EPOC CON EXACERBACION	99	45
6	Barcelona	166	13/7/18 22:50	90	1	2	41,38112102	2,131493157	1	DISNEA	99	90
7	Barcelona	172	1/6/18 13:36	92	1	2	41,39992096	2,132714049	1	DISNEA	99	60
8	Barcelona	161	27/7/18 9:35	83	1	2	41,41477019	2,148168477	1	FIEBRE	99	45
9	Barcelona	167	16/1/18 10:31	67	1	2	41,40105862	2,121924827	1	PNEUMONIA	99	60

*Sorting by numerical column **Patient Age** - 2*



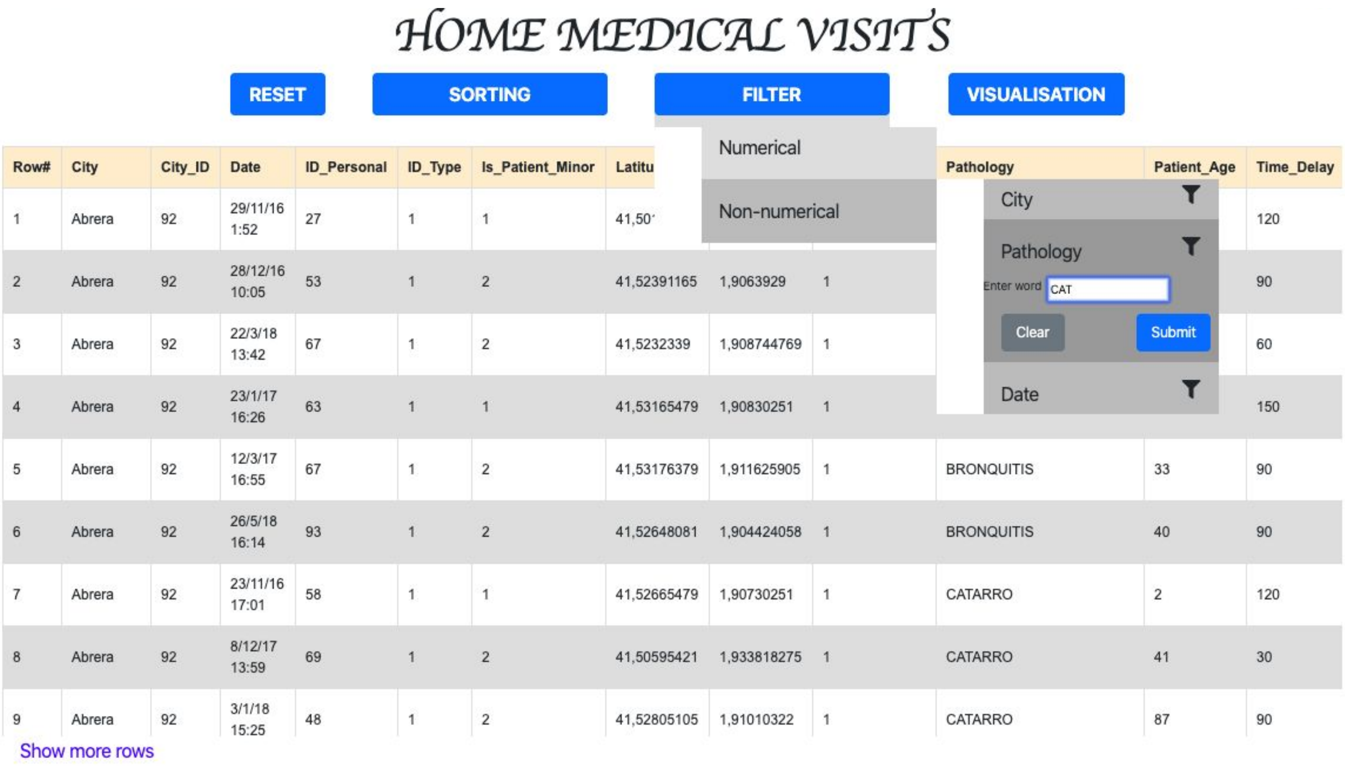
HOME MEDICAL VISITS

10	Barcelona	144	12/3/18 14:21	66	1	2	41,40917077	2,153666889	1	TOS	99	90
11	Bellaterra	3	18/12/17 13:42	80	1	2	41,51278077	2,098593341	1	CURAS EN GENERAL -	99	60
12	Bigues i Riells	126	4/8/18 8:10	55	1	2	41,68644	2,215529108	1	FIEBRE	99	60
13	Caldes de Montbui	53	10/10/17 0:18	70	1	2	41,63153645	2,171203572	1	DISNEA	99	60
14	Castellar del Vallaaaas	4	25/6/18 9:55	84	1	2	41,6217983	2,095415403	1	ASTENIA	99	120
15	Cerdanyola	6	26/4/16 16:25	51	1	2	41,49792196	2,145698077	1	CELULITIS	99	0
16	Cerdanyola	6	1/6/16 12:56	51	1	2	41,49992196	2,150698077	1	DOLOR ARTICULAR	99	45
17	Cornellaaaa de Llobregat	99	29/11/16 12:29	48	1	2	41,37149207	2,085386607	1	DISNEA	99	45
18	Cornellaaaa de Llobregat	99	10/1/17 19:07	61	1	2	41,37449207	2,089386607	1	INCONTINENCIA URINARIA	99	120
	Hospitalet de		16/1/18									

Sorting by numerical column ***Patient Age*** *- 3*

c.) Filtering on a non-numerical column

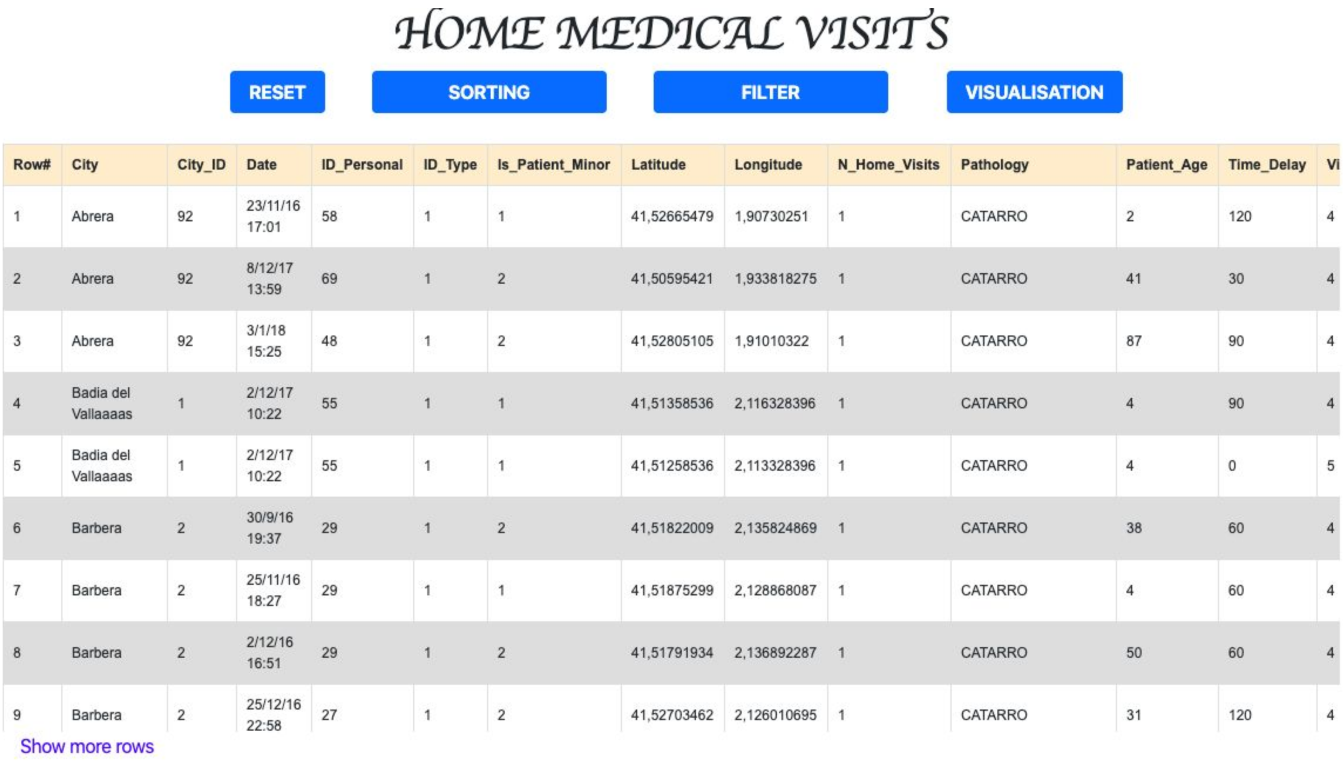
The data in the table, column name and the string on which the data needs to be filtered is converted to a JSON object and send to the Flask API point for filtering. It is then converted to a dataframe in Pandas on which filtering operation is performed. The filtered dataframe is converted back to a JSON object which is then displayed on the HTML page through Javascript.



Enter word CAT

Row#	City	City_ID	Date	ID_Personal	ID_Type	Is_Patient_Minor	Latitu			Pathology	Patient_Age	Time_Delay
1	Abrera	92	29/11/16 1:52	27	1	1	41,50					120
2	Abrera	92	28/12/16 10:05	53	1	2	41,52391165	1,9063929	1			90
3	Abrera	92	22/3/18 13:42	67	1	2	41,5232339	1,908744769	1			60
4	Abrera	92	23/1/17 16:26	63	1	1	41,53165479	1,90830251	1			150
5	Abrera	92	12/3/17 16:55	67	1	2	41,53176379	1,911625905	1	BRONQUITIS	33	90
6	Abrera	92	26/5/18 16:14	93	1	2	41,52648081	1,904424058	1	BRONQUITIS	40	90
7	Abrera	92	23/11/16 17:01	58	1	1	41,52665479	1,90730251	1	CATARRO	2	120
8	Abrera	92	8/12/17 13:59	69	1	2	41,50595421	1,933818275	1	CATARRO	41	30
9	Abrera	92	3/1/18 15:25	48	1	2	41,52805105	1,91010322	1	CATARRO	87	90

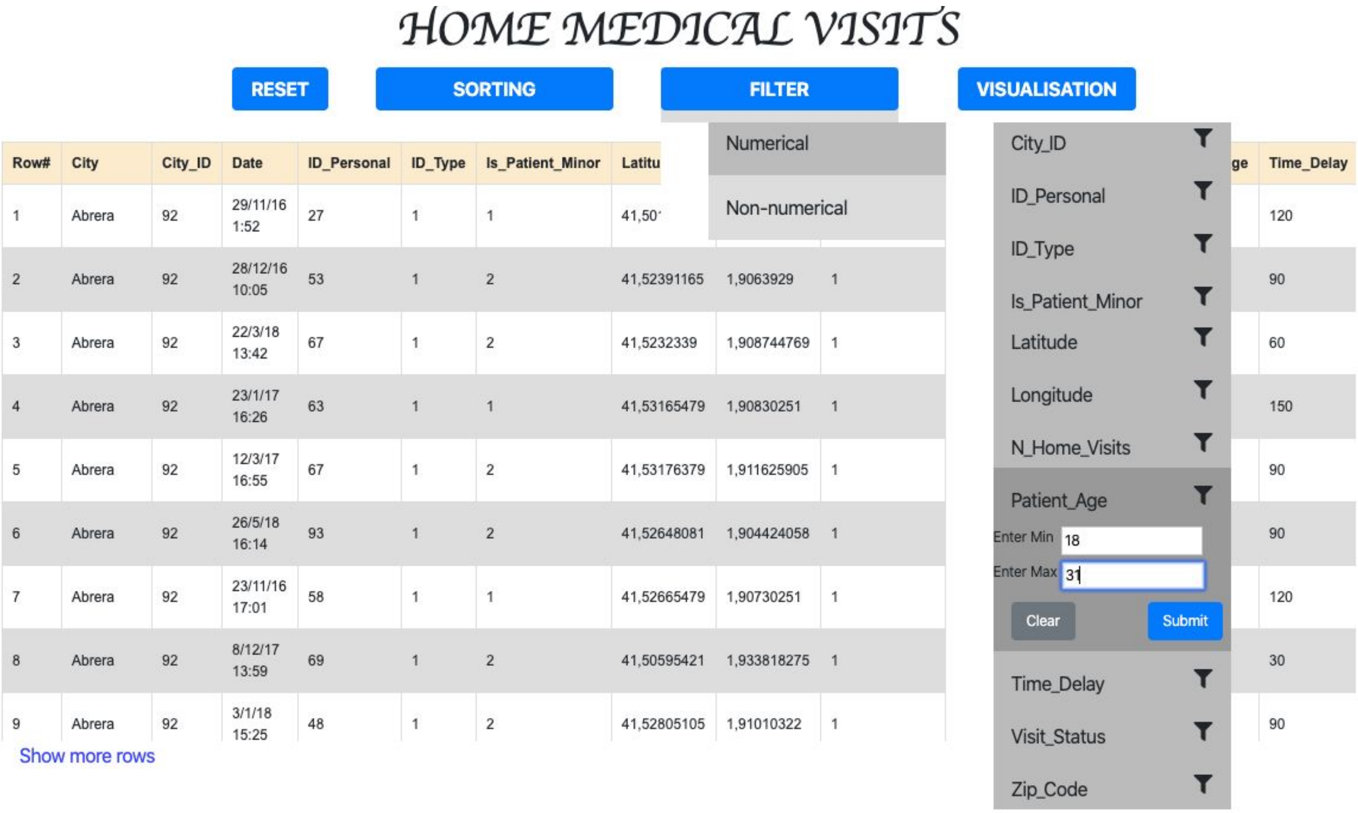
Filtering on non-numerical column ***Pathology*** *- 1*



Row#	City	City_ID	Date	ID_Personal	ID_Type	Is_Patient_Minor	Latitude	Longitude	N_Home_Visits	Pathology	Patient_Age	Time_Delay	Vi
1	Abrera	92	23/11/16 17:01	58	1	1	41,52665479	1,90730251	1	CATARRO	2	120	4
2	Abrera	92	8/12/17 13:59	69	1	2	41,50595421	1,933818275	1	CATARRO	41	30	4
3	Abrera	92	3/1/18 15:25	48	1	2	41,52805105	1,91010322	1	CATARRO	87	90	4
4	Badia del Vallaaaas	1	2/12/17 10:22	55	1	1	41,51358536	2,116328396	1	CATARRO	4	90	4
5	Badia del Vallaaaas	1	2/12/17 10:22	55	1	1	41,51258536	2,113328396	1	CATARRO	4	0	5
6	Barbera	2	30/9/16 19:37	29	1	2	41,51822009	2,135824869	1	CATARRO	38	60	4
7	Barbera	2	25/11/16 18:27	29	1	1	41,51875299	2,128868087	1	CATARRO	4	60	4
8	Barbera	2	2/12/16 16:51	29	1	2	41,51791934	2,136892287	1	CATARRO	50	60	4
9	Barbera	2	25/12/16 22:58	27	1	2	41,52703462	2,126010695	1	CATARRO	31	120	4

*Filtering on non-numerical column **Pathology** - 2* d.) Filtering on a numerical column

The data in the table, column name, a range of max and min value on which the data needs to be filtered is converted to a JSON object and send to the Flask API point for filtering. It is then converted to a dataframe in Pandas on which filtering operation is performed. The filtered dataframe is converted back to a JSON object which is then displayed on the HTML page through Javascript.

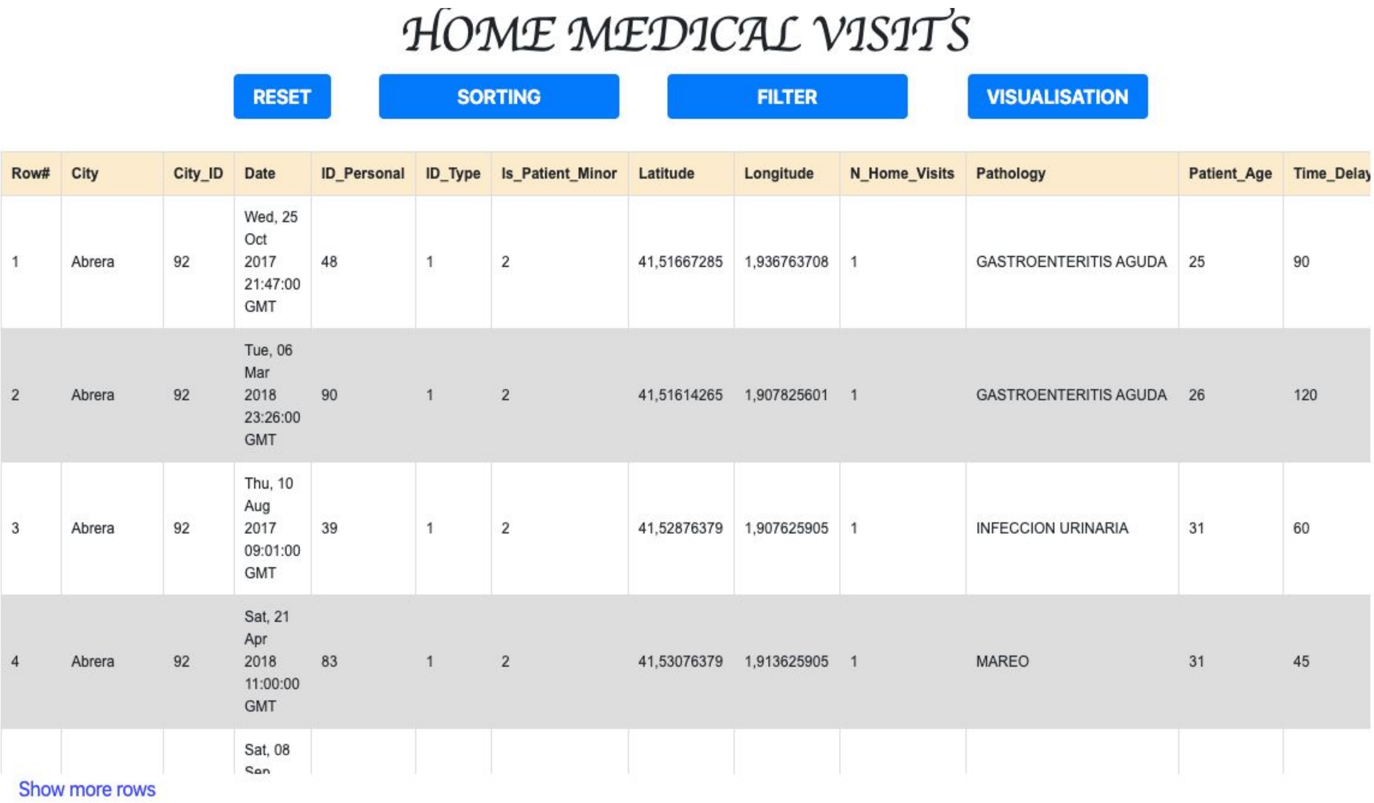


Enter Min 18

Enter Max 31

Row#	City	City_ID	Date	ID_Personal	ID_Type	Is_Patient_Minor	Latitu			ge	Time_Delay
1	Abrera	92	29/11/16 1:52	27	1	1	41,50'				120
2	Abrera	92	28/12/16 10:05	53	1	2	41,52391165	1,9063929	1		90
3	Abrera	92	22/3/18 13:42	67	1	2	41,5232339	1,908744769	1		60
4	Abrera	92	23/1/17 16:26	63	1	1	41,53165479	1,90830251	1		150
5	Abrera	92	12/3/17 16:55	67	1	2	41,53176379	1,911625905	1		90
6	Abrera	92	26/5/18 16:14	93	1	2	41,52648081	1,904424058	1		90
7	Abrera	92	23/11/16 17:01	58	1	1	41,52665479	1,90730251	1		120
8	Abrera	92	8/12/17 13:59	69	1	2	41,50595421	1,933818275	1		30
9	Abrera	92	3/1/18 15:25	48	1	2	41,52805105	1,91010322	1		90

*Filtering on numerical column **Patient Age** - 1*

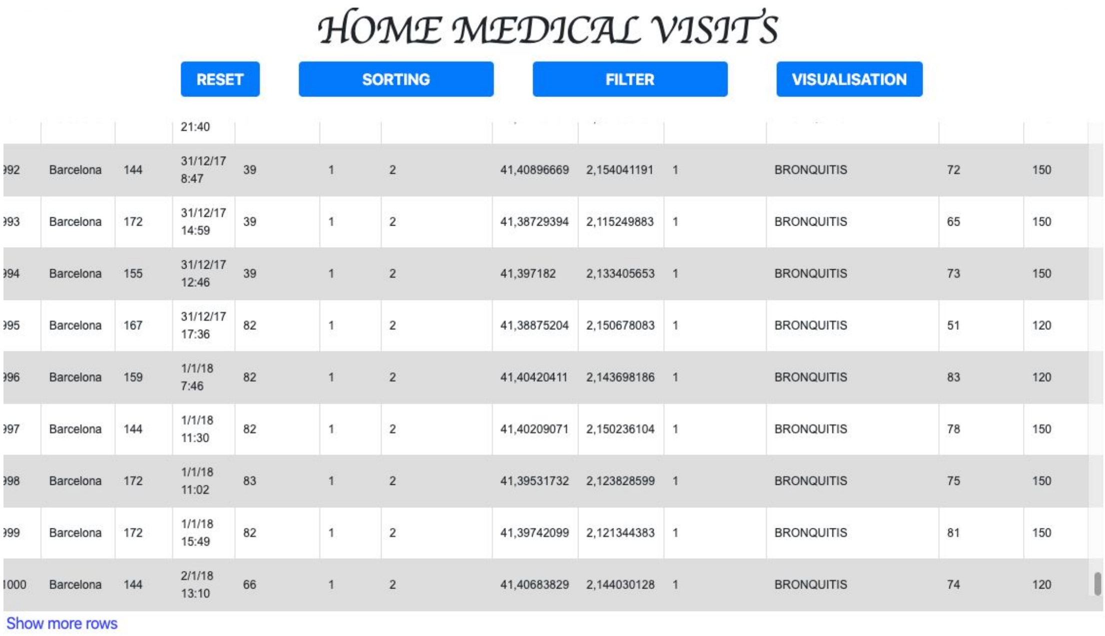


Row#	City	City_ID	Date	ID_Personal	ID_Type	Is_Patient_Minor	Latitude	Longitude	N_Home_Visits	Pathology	Patient_Age	Time_Delay
1	Abrera	92	Wed, 25 Oct 2017 21:47:00 GMT	48	1	2	41,51667285	1,936763708	1	GASTROENTERITIS AGUDA	25	90
2	Abrera	92	Tue, 06 Mar 2018 23:26:00 GMT	90	1	2	41,51614265	1,907825601	1	GASTROENTERITIS AGUDA	26	120
3	Abrera	92	Thu, 10 Aug 2017 09:01:00 GMT	39	1	2	41,52876379	1,907625905	1	INFECCION URINARIA	31	60
4	Abrera	92	Sat, 21 Apr 2018 11:00:00 GMT	83	1	2	41,53076379	1,913625905	1	MAREO	31	45
			Sat, 08 Sep									

*Filtering on numerical column **Patient Age** - 2*

e.) Pagination

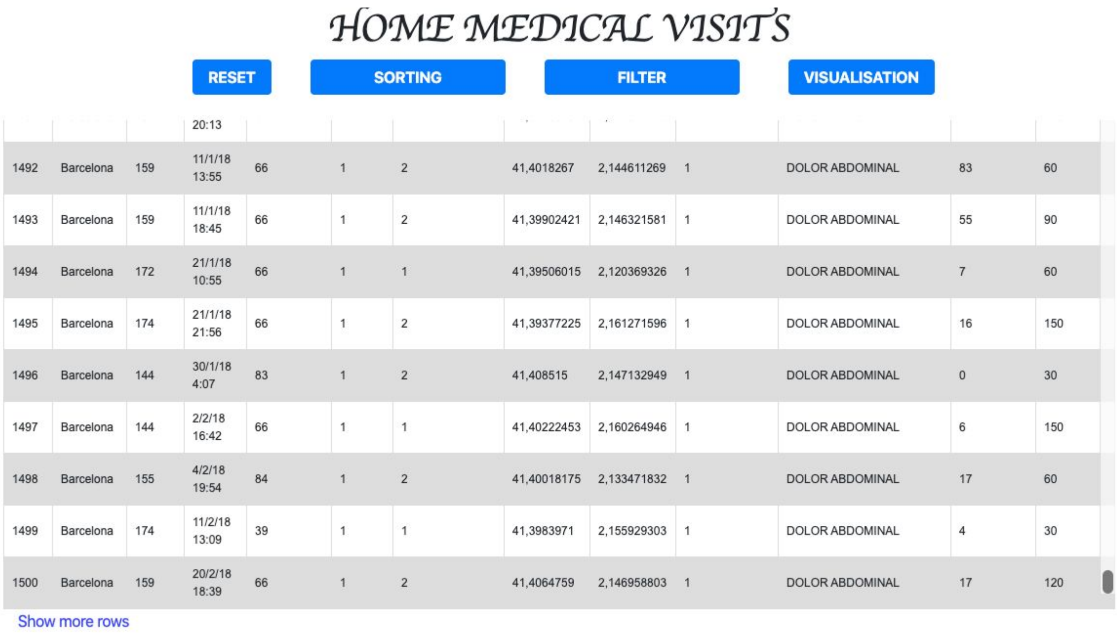
We designed and implemented pagination to speed up the running time. The initial table only displays 1000 rows. When user click "**Show more rows**", the table displays the next 500 rows. This improves user experience a lot because it takes a while to display all data.



			21:40									
992	Barcelona	144	31/12/17 8:47	39	1	2	41,40896669	2,154041191	1	BRONQUITIS	72	150
993	Barcelona	172	31/12/17 14:59	39	1	2	41,38729394	2,115249883	1	BRONQUITIS	65	150
994	Barcelona	155	31/12/17 12:46	39	1	2	41,397182	2,133405653	1	BRONQUITIS	73	150
995	Barcelona	167	31/12/17 17:36	82	1	2	41,38875204	2,150678083	1	BRONQUITIS	51	120
996	Barcelona	159	1/1/18 7:46	82	1	2	41,40420411	2,143698186	1	BRONQUITIS	83	120
997	Barcelona	144	1/1/18 11:30	82	1	2	41,40209071	2,150236104	1	BRONQUITIS	78	150
998	Barcelona	172	1/1/18 11:02	83	1	2	41,39531732	2,123828599	1	BRONQUITIS	75	150
999	Barcelona	172	1/1/18 15:49	82	1	2	41,39742099	2,121344383	1	BRONQUITIS	81	150
1000	Barcelona	144	2/1/18 13:10	66	1	2	41,40683829	2,144030128	1	BRONQUITIS	74	120

Show more rows

Pagination with no click - 1



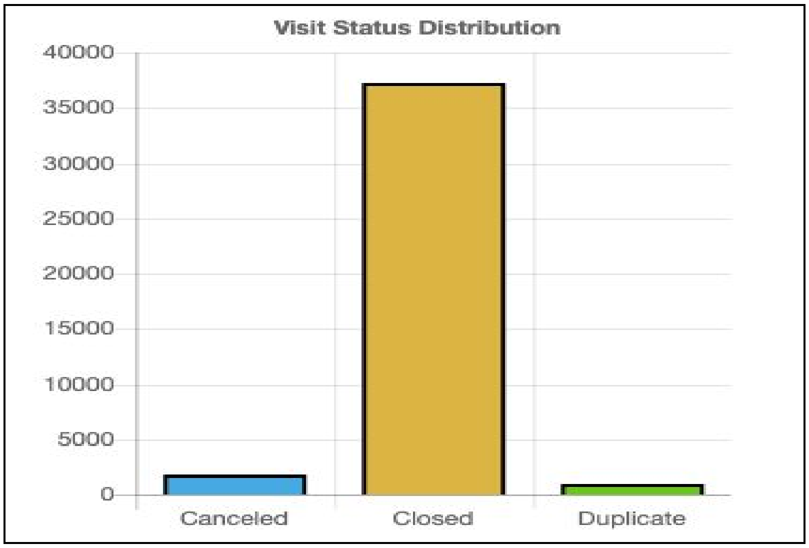
			20:13									
1492	Barcelona	159	11/1/18 13:55	66	1	2	41,4018267	2,144611269	1	DOLOR ABDOMINAL	83	60
1493	Barcelona	159	11/1/18 18:45	66	1	2	41,39902421	2,146321581	1	DOLOR ABDOMINAL	55	90
1494	Barcelona	172	21/1/18 10:55	66	1	1	41,39506015	2,120369326	1	DOLOR ABDOMINAL	7	60
1495	Barcelona	174	21/1/18 21:56	66	1	2	41,39377225	2,161271596	1	DOLOR ABDOMINAL	16	150
1496	Barcelona	144	30/1/18 4:07	83	1	2	41,408515	2,147132949	1	DOLOR ABDOMINAL	0	30
1497	Barcelona	144	2/2/18 16:42	66	1	1	41,40222453	2,160264946	1	DOLOR ABDOMINAL	6	150
1498	Barcelona	155	4/2/18 19:54	84	1	2	41,40018175	2,133471832	1	DOLOR ABDOMINAL	17	60
1499	Barcelona	174	11/2/18 13:09	39	1	1	41,3983971	2,155929303	1	DOLOR ABDOMINAL	4	30
1500	Barcelona	159	20/2/18 18:39	66	1	2	41,4064759	2,146958803	1	DOLOR ABDOMINAL	17	120

Show more rows

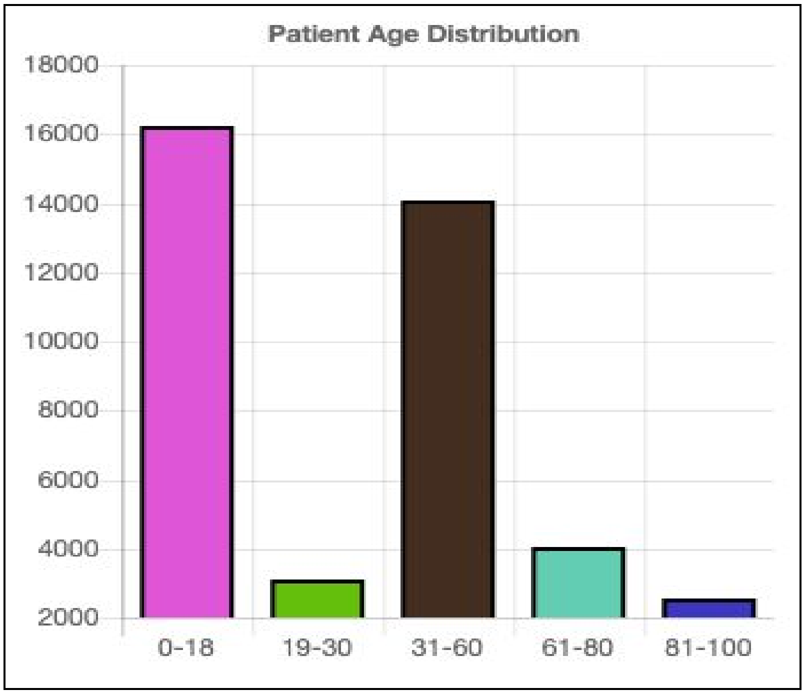
Pagination with one click - 2

f.) Data Visualisation

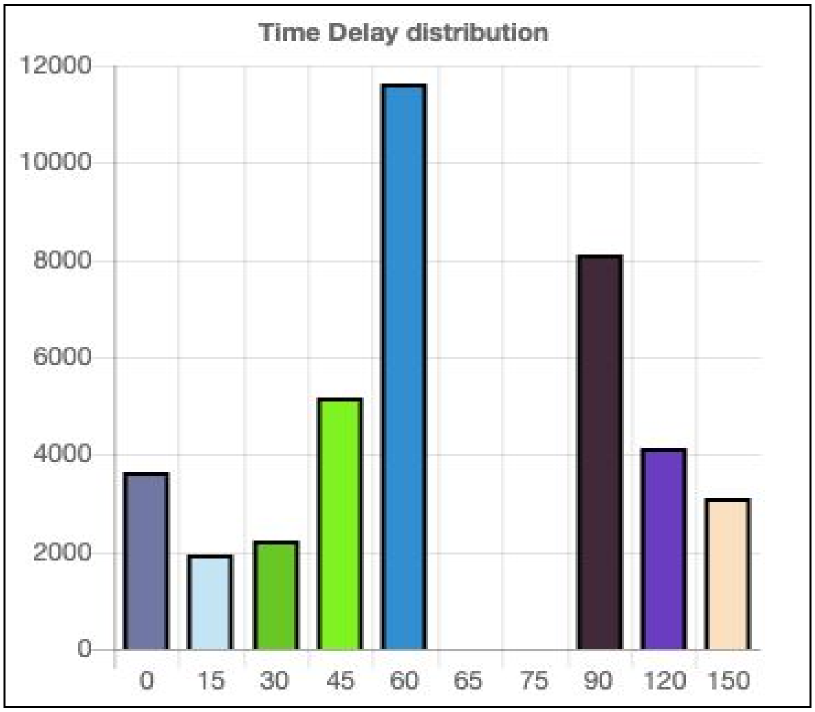
Following graphs give some descriptive analysis of the entire data (irrespective of whether data is sorted or filtered)



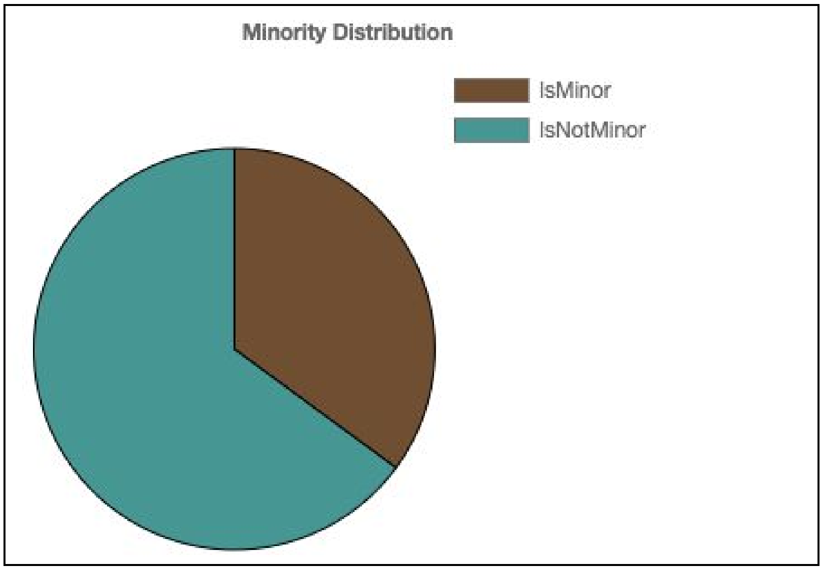
Graph 1 - Gives the count for different visit status categories. This shows that most of the visits have been completed and closed.



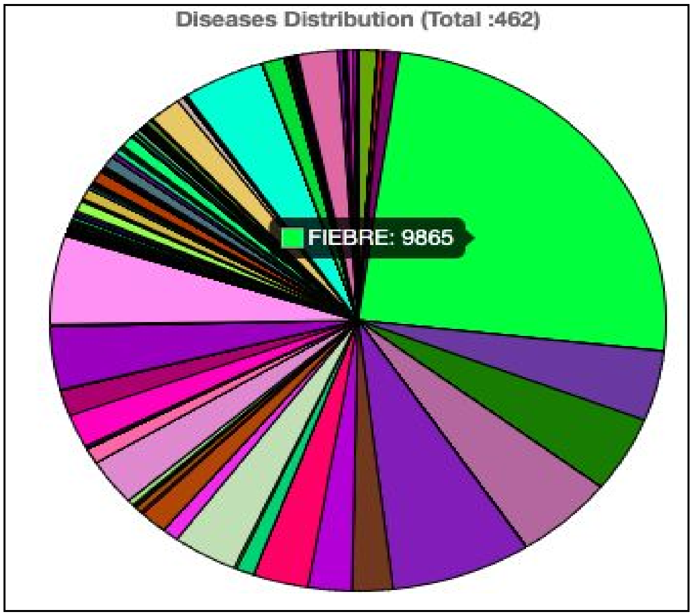
Graph 2 - Gives the distribution of age groups of patients in the dataset. 16000 patients in the dataset belong to the age group of 0-18.



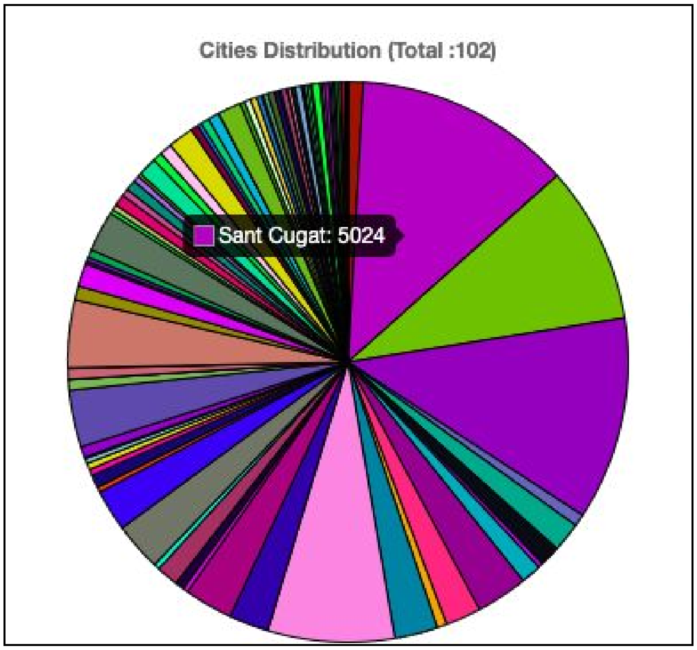
Graph 3 - Gives the distribution of the time delay in visits given by the Home Medical Services at different households.



Graph 4 - Gives minority distribution among patients. The percentage of non-minority is greater than that of minority in the dataset.



Graph 5 - Gives distribution of diseases among patients. The disease FIEBRE is prominent in the dataset.



Graph 6 - Gives the distribution of cities from which the records are obtained. Maximum records are obtained from Sant Cugat.

Responsibility and work done

Keerti Bhogaraju
1.) Fetching the dataset from firebase and displaying it as an HTML table.
2.) Filtering operation on numerical and non-numerical columns.
3.) Performing visualization on the entire dataset

Kevin Chiang
1.) Uploading the dataset from CSV format to the Firebase database.
2.) Sorting operation on numerical and non-numerical columns.
3.) Implementing pagination function

Future Scope

1.) Display descriptive statistics for variables
2.) Importing generic dataset and performing functionalities on the same
3.) Exporting results in different formats
4.) Advanced visualization